UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/730,956	12/09/2003	Mark-Graham Stoodley	CA920030008US1	3689

7590 12/19/2006

Marilyn Smith Dawkins
International Business Machines
Intellectual Property Law
11400 Burnet Road
Austin, TX 78758

EXAMINER	
YAARY, MICHAEL D	
ART UNIT	PAPER NUMBER
2196	

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	12/19/2006	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

Office Action Summary	Application No.	Applicant(s)	
	10/730,956	STOODLEY, MARK-GRAHAM	
	Examiner	Art Unit	
	Michael Yaary	2196	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *09 December 2003*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-10* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-10* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *09 December 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *12/09/2003*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

Detailed Action

1. Claims 1-10 are pending in the application.

Claim Objections

2. Claim 1, line 6; claim 3, line 6; and claim 4, line 7; recite the limitation "said lock." There is insufficient antecedent basis for this limitation in the claim.

3. Claim 5, line 4, recites the limitation "said each instruction." There is insufficient antecedent basis for this limitation in the claim.

Claim Rejections - 35 USC § 101

4. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 3 is rejected under U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. The claim is directed to an "exception handler," which appears to be software with functional descriptive material. Thus, being ineligible for protection.

Claim 9 is rejected under U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. The claims are directed to a "compiler" which appears to be functional descriptive material, software per se. Thus, being ineligible for protection.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 1, 3, and 4 are rejected under 35 U.S.C. 103(a) as being unpatentable over Sokolov (hereafter Sokolov(1))(US Pub. 2003/0079202) in view of Sokolov (hereafter Sokolov(2))(US Pub. 2003/0079203)

Regarding claim 1, Sokolov(1) discloses a method of handling an exception (abstract, lines 1-2) comprising:

Recognizing that an exception has occurred during the execution of a given method ([0015], lines 1-5);

Consulting a stack frame associated with said given method ([0024], lines 5-12).

Sokolov(1), however, does not teach determining the identity of an object required to be unlocked and removing said lock on said object.

Sokolov(2) does teach determining the identity of an object required to be unlocked ([0015], lines 1-18 disclose the method of executing a synchronized Java method. In the method monitors (locks) are referenced to their association with Java methods and are released based on the appropriate reference that has been determined.) and removing said lock on said object ([0015], lines 15-17).

Therefore, it would have been obvious to one of ordinary skill in the are at the time of the invention to modify Sokolov(1) by determining the identity of an object required to be unlocked and removing said lock on said object as taught by Sokolov(2), in order to efficiently execute JAVA methods during an exception occurrence.

Regarding claim 3, Sokolov(1) discloses an exception handler (abstract, lines 6-7) operable to:

Recognizing that an exception has occurred during the execution of a given method ([0015], lines 1-5);

Consulting a stack frame associated with said given method ([0024], lines 5-12).

Sokolov(1), however, does not teach determining the identity of an object required to be unlocked and removing said lock on said object.

Sokolov(2) does teach determining the identity of an object required to be unlocked ([0015], lines 1-18 disclose the method of executing a synchronized Java method. In the method monitors (locks) are referenced to their association with Java methods and are released based on the appropriate reference that has been determined.) and removing said lock on said object ([0015], lines15-17).

Therefore, it would have been obvious to one of ordinary skill in the are at the time of the invention to modify Sokolov(1) by determining the identity of an object required to be unlocked and removing said lock on said object as taught by Sokolov(2), in order to efficiently execute JAVA methods during an exception occurrence.

Regarding claim 4, Sokolov(1) discloses a computer readable medium containing computer-executable instructions ([0013], lines 1-3) which, when performed by a processor in a computer system, cause said computer system to:

Recognizing that an exception has occurred during the execution of a given method ([0015], lines 1-5);

Consulting a stack frame associated with said given method ([0024], lines 5-12).

Sokolov(1), however, does not teach determining the identity of an object required to be unlocked and removing said lock on said object.

Sokolov(2) does teach determining the identity of an object required to be unlocked ([0015], lines 1-18 disclose the method of executing a synchronized Java method. In the method monitors (locks) are referenced to their association with Java methods and are released based on the appropriate reference that has been determined.) and removing said lock on said object ([0015], lines15-17).

Therefore, it would have been obvious to one of ordinary skill in the are at the time of the invention to modify Sokolov(1) by determining the identity of an object required to be unlocked and removing said lock on said object as taught by Sokolov(2), in order to efficiently execute JAVA methods during an exception occurrence.

7. Claims 5-10 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bak et al. (hereafter Bak)(US Pat. 6,167,424) in view of Cohn et al. (hereafter Cohn)(US Pat. 5,901,308).

Regarding claim 5, Bak discloses at a just in time compiler of programming language code, said programming language code including a plurality of instructions, a method of generating executable code (column 24, lines 61-66) comprising:

Determining a synchronization depth for said each instruction (column 3, lines 14-29 and column 4, lines 14-25 disclose the method of which thread synchronization is determined based on maintaining a counter used for determining the amount of times a thread has locked and unlocked an object.);

Bak does not disclose that the method comprises:

Associating said synchronization depth with a program counter address associated with said each instruction;

Determining a continuous range of program counter addresses associated with an equivalent synchronization depth; and

Storing an indication of said continuous range of program counter addresses in a table associated with said synchronization depth.

However, Cohn discloses that the method comprises:

Associating said synchronization depth with a program counter address associated with said each instruction (column 7, lines 18-20 and column 7, lines 37-40);

Determining a continuous range of program counter addresses associated with an equivalent synchronization depth (column 7, lines 37-45); and

Storing an indication of said continuous range of program counter addresses in a table associated with said synchronization depth (Column 3, lines 28-41 disclose a table used to store a range of address locations identifying instructions causing exceptions.).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Bak by associating said synchronization depth with a program counter address, determining a continuous range of program counter addresses associated with an equivalent synchronization depth, and storing an indication of said continuous range of program counter addresses in a table as taught by Cohn, for the benefit of efficiently executing instructions during the occurrence of an exception.

Regarding claim 9, Bak discloses a just in time compiler of programming language code, said programming language code including a plurality of instructions, said compiler operable to (column 24, lines 61-66):

Determining a synchronization depth for said each instruction (column 3, lines 14-29 and column 4, lines 14-25 disclose the method of which thread synchronization is determined based on a maintaining a counter used for determining the amount of times a thread has locked and unlocked an object.);

Bak does not disclose that the method comprises:

Associating said synchronization depth with a program counter address associated with said each instruction;

Determining a continuous range of program counter addresses associated with an equivalent synchronization depth; and

Storing an indication of said continuous range of program counter addresses in a table associated with said synchronization depth.

However, Cohn discloses that the method comprises:

Associating said synchronization depth with a program counter address associated with said each instruction (column 7, lines 18-20 and column 7, lines 37-40);

Determining a continuous range of program counter addresses associated with an equivalent synchronization depth (column 7, lines 37-45); and

Storing an indication of said continuous range of program counter addresses in a table associated with said synchronization depth (Column 3, lines 28-41 disclose a table used to store a range of address locations identifying instructions causing exceptions.).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Bak by associating said synchronization depth with a program counter address, determining a continuous range of program counter addresses associated with an equivalent synchronization depth, and storing an indication of said continuous range of program counter addresses in a table as taught by Cohn, for the benefit of efficiently executing instructions during the occurrence of an exception.

Regarding claim 10, Bak discloses a computer readable medium containing computer executable instructions which, when performed by a processor in a computer system, cause said computer system to (abstract) :

Determining a synchronization depth for said each instruction (column 3, lines 14-29 and column 4, lines 14-25 disclose the method of which thread synchronization is determined based on a maintaining a counter used for determining the amount of times a thread has locked and unlocked an object.);

Bak does not disclose that the method comprises:

Associating said synchronization depth with a program counter address associated with said each instruction;

Determining a continuous range of program counter addresses associated with an equivalent synchronization depth; and

Storing an indication of said continuous range of program counter addresses in a table associated with said synchronization depth.

However, Cohn discloses that the method comprises:

Associating said synchronization depth with a program counter address associated with said each instruction (column 7, lines 18-20 and column 7, lines 37-40);

Determining a continuous range of program counter addresses associated with an equivalent synchronization depth (column 7, lines 37-45); and

Storing an indication of said continuous range of program counter addresses in a table associated with said synchronization depth (Column 3, lines 28-41 disclose a table used to store a range of address locations identifying instructions causing exceptions.).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Bak by associating said synchronization depth with a program counter address, determining a continuous range of program counter addresses associated with an equivalent synchronization depth, and storing an indication of said continuous range of program counter addresses in a table as taught by Cohn, for the benefit of efficiently executing instructions during the occurrence of an exception.

Regarding claim 6, Bak further discloses said programming language code is Java (column 13, lines 60-63).

Regarding claim 7, Bak does not disclose determining said continuous range of program counter addresses comprises:

Where a first synchronization depth determined for a first instruction differs from a second synchronization depth determined for a directly preceding instruction,

Storing said synchronization depth in said table associated with a given range of program counter addresses,

Where said given range of program counter addresses includes said program counter address of said directly preceding instruction.

However, Cohn discloses determining said continuous range of program counter addresses comprises:

Where a first synchronization depth determined for a first instruction differs from a second synchronization depth determined for a directly preceding instruction (column 7, lines 6-10 and lines 29-36),

Storing said synchronization depth in said table associated with a given range of program counter addresses (Column 3, lines 28-41 disclose a table used to store a range of address locations identifying instructions causing exceptions.) ,

Where said given range of program counter addresses includes said program counter address of said directly preceding instruction (Inherent in column 3, lines 28-41 as if the instructions causing exceptions are identified in the table, then the range of addresses will include an address of a directly preceding instruction).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Bak by having a first synchronization depth determined for a first instruction differs from a second synchronization depth determined for a directly preceding instruction, storing said synchronization depth in said table associated with a given range of program counter addresses, and said given range of program counter addresses includes said program counter address of said directly preceding instruction as taught by Cohn, for the benefit of efficiently executing instructions during the occurrence of an exception.

Regarding claim 8, Bak does not disclose said plurality of instructions may be arranged in a plurality of basic blocks of code, further comprising determining a synchronization depth at the beginning of each said basic block of code.

However, Cohn discloses said plurality of instructions may be arranged in a plurality of basic blocks of code, further comprising determining a synchronization depth at the beginning of each said basic block of code (column 2, lines 8-16).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Bak by having said plurality of instructions may be arranged in a plurality of basic blocks of code, further comprising determining a synchronization depth at the beginning of each said basic block of code as taught by Cohn for the benefit of efficiently executing instructions during the occurrence of an exception.

8. Claim 2 is rejected under 35 U.S.C. 103(a) as being unpatentable over Sokolov(1) in view of Sokolov(2) as applied to claim 1 above, and further in view of Yoshioka et al. (hereafter Yoshioka)(US Pat. 6,425,039).

Regarding claim 2, Sokolov(1) and Sokolov(2) do not disclose determining a program counter address at which said exception has occurred;

Determining, from a table, a synchronization depth associated with said program counter address; and

Wherein said consulting said stack frame includes reading from a location associated with said synchronization depth.

However, Yoshioka discloses determining a program counter address at which said exception has occurred (column 3, lines 13-18);

Determining, from a table, a synchronization depth associated with said program counter address (column 14, lines 4-13); and

Wherein said consulting said stack frame includes reading from a location associated with said synchronization depth (column 2, lines 12-18).

Therefore, it would have been obvious to one of ordinary skill in the are at the time of the invention to modify Sokolov(1) and Sokolov(2) by, determining a program counter address at which an exception has occurred, Determining, from a table, a synchronization depth associated with the program counter address, and when consulting the stack frame reading from a location associated with the synchronization depth as taught by Yoshioka, for the benefit of reducing execution time when handling exceptions.

Conclusions

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael Yaary whose telephone number is (571) 270-1249. The examiner can normally be reached on Monday-Friday, 8:00 a.m - 5:00 p.m..

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Nabil El-Hady can be reached on (571) 272-3963. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

MY

MY

N. El-Hady

NABIL M. EL-HADY
SUPERVISORY PATENT EXAMINER